

屏幕取词项目移交

1.背景

屏幕取词的基本功能是实现点击屏幕区域，获得相应区域的文本内容，具体实现可以参照金山词霸的屏幕取词功能。



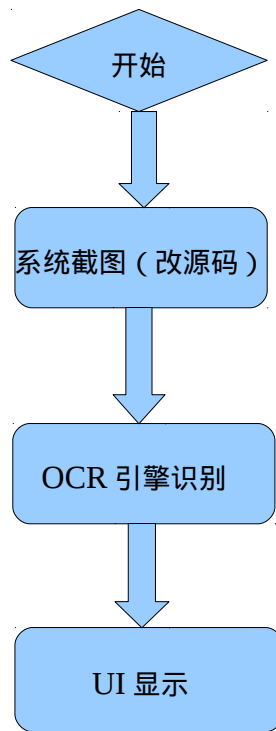
1-1 金山词霸取词

本项目的初期版本已经实现了基本的取词功能，效果图如下图。



1-2 本项目取词效果

具体实现步骤，分三步。流程图如下图。



1-3 流程图

2.具体实现

2.1 系统截图

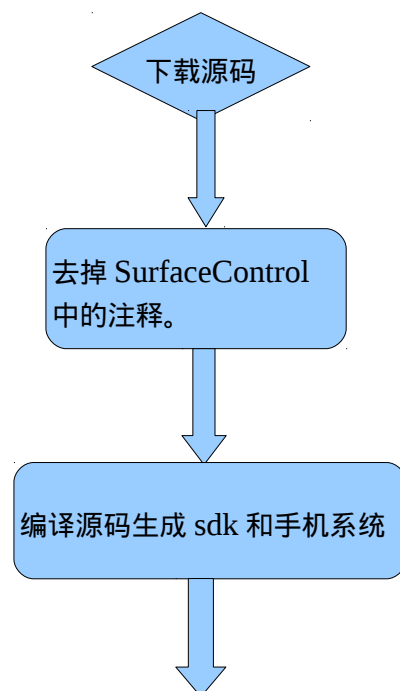
具体实现方式是可以截取 android 系统任意界面的视图，包括状态栏等。先后尝试了六种方法，分别是:bufferframe 读取 fb0，反射方法，编译源码去掉 “@hide”，加入自己编译的 core 文件和 framework 文件，googlecode 开源项目 asl，手机端调用 adb shell 中的截屏命令。（具体失败原因可参见我的 blog：<http://blog.csdn.net/buptgshengod/article/details/20607281>）

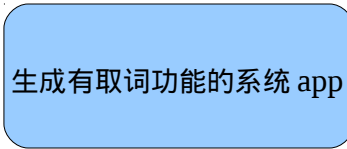
最后通过更改源码 SurfaceControl.java 文件成功实现了截屏功能。

长按 android 手机的电源键与音量减键可以实现系统截屏，通过探索源码，找到了这个功能的具体实现就是通过 SurfaceControl.java（探索过程见：

<http://blog.csdn.net/buptgshengod/article/details/19911909>）。

实现这个功能的具体思路是如下流程图：





生成有取词功能的系统 app

2.1-1 截屏流程图

2.1.1 下载源码

本项目对应的事 tianchi-dsds 版本的源码，对应的下载地址：

https://wiki.sonyericsson.net/androiki/ASW_CM/yukon-dsds。

下载源码需要先开通 git 权限（见新员工手册），具体命令

```
$ repo init -u git://review.sonyericsson.net/platform/manifest -b jb-mr2-yukon-dsds  
$ repo sync
```

2.1.2 去掉 surfacecontrol 中的注释

下载完成后替换/frameworks/base/core/java/android/view 中的 SurfaceControl 文件为项目附件中的 SurfaceControl 文件即可。因为源码中原始的 surfacecontrol 函数是加 “@hide” 的，在 sdk 层不可见，我们去掉这个约束编译自己的 sdk 与系统，就可以调用 surfacecontrol 中的截屏函数。

2.1.3 生成 sdk 和手机系统

生成 sdk 的命令（在源码目录下）：

```
$ source build/envsetup.sh
```

```
$ lunch tianchi_dsds-eng
```

```
$ make -j5 PRODUCT-semcsdk-semcsdk
```

执行完后 sdk 将在源码目录的 `/out/host/linux-x86/sdk/` 路径下。

生成手机系统（在源码目录下）：

```
$ source build/envsetup.sh
```

```
$ lunch tianchi_dsds-eng
```

```
$ make -j10 fullbuild
```

软件将

`out/target/product/tianchi_dsds/obj/semcpkgtoimg_staging/*.zip`

包烧入手机，手机将配置有自己生成的系统。

2.1.4 生成系统 app

因为为了实现截屏功能已经修改了 sdk 以及手机系统，所以要求生成的 app 拥有 system 权限才可以实现截屏功能。具体方法是

（1）在 app 的 `AndroidManifest.xml` 中加入
`android:sharedUserId="android.uid.system"`

在 `eclipse` 直接编译会报错，因为 system app 必须在源码中编译。

(2) 把工程放到源码的/**packages/apps** 目录下
在 **terminal** 中 **cd** 到对应目录输入：
semc-build debug (调试工程)
semc-build install (安装工程到手机中)

2.2 OCR 引擎

使用的是 google 的开源 ocr 引擎 tesseract。我已经编译好了相关 jni 文件。工程中的 libs 就是导入的编译好的 OCR 引擎。

(demon 使用可以参见：

<http://blog.csdn.net/buptgshengod/article/details/18810053>)

在工程中的使用：

```
baseApi=new TessBaseAPI();
```

```
baseApi.init("/mnt/sdcard/tesseract/",  
"chi_sim");
```

语言包下载地址：http://code.google.com/p/tesseract-ocr/downloads/detail?name=chi_sim.traineddata.gz&can=2&q=

下载后放到手机sdcard/tesseract/tessdata目录下即可被本工程调用。本工程使用的tesseract版本是3.0.1版。

3. 附加功能

3.1 多手势功能

悬浮窗的大小可以通过两个手指的伸拉来控制，代码实现在 FxService.java 中的

```
        if (event.getPointerCount() == 2){
            Toast.makeText(FxService.this, "haha",
Toast.LENGTH_SHORT).show();

            float x = event.getX(0) - event.getX(1);
            float y = event.getY(0) - event.getY(1);
            float value = (float) Math.sqrt(x * x + y * y); // 计算两点的
距离

            if (baseValue == 0) {
                baseValue = value;
            }
            else {
                if (value - baseValue >= 10) {
                    Bwidth=Bwidth+3;
                    Bheight=Bheight+3;
                }
                if (value - baseValue <= -10) {
                    Bwidth=Bwidth-3;
                    Bheight=Bheight-3;
                }
            }
            mFloatView.setWidth(Bwidth);
            mFloatView.setHeight(Bheight);
        }
```

3.2 数据库

代码中的 dbHelper.java 实现具体功能。所取到过的词都可以在进入界面中显示，如下图



(demon
使用可以

参考： <http://blog.csdn.net/buptgshengod/article/details/22273265#reply>)

4. 代码注释

FloatingWindow 包括三个类，
MainActivity, FxService, dbHelper。

4.1 MainActivity

在 onCreate 里设置数据库显示的 listview，通过按键启动屏幕取词 service，这个 service 在类 FxService 中定义。

4.2 FxService

在 creat 中初始化悬浮窗以及相应的显示窗，初始化数据库和 OCR 引擎。在 mFloatView 的 ACTION_DOWN 监听中 handler.postDelayed(runnable, 1000)，表示当点击悬浮窗超过一秒钟启动 runnable。在 runnable 中完成了截图功能以及 OCR 引擎的识别。截屏功能因为我们更改了源码，生成的自己的 sdk 中只要
[bp=SurfaceControl.screenshot\(width, length\);](#)
一句话就实现了截屏。

每次截屏完，然后截取手指点击的部分区域图片
[Bitmap saveBitmap =](#)
[Bitmap.createBitmap\(getDiskBitmap\("/sdcard/Pictures/ScreenShot.png"\), wmParams.x-15,](#)
[wmParams.y-10 , 90, 50 \);](#)

将截取的图片送往 OCR 引擎识别，返回 text 信息。

`baseApi.setImage(saveBitmap);`

`String text1= baseApi.getUTF8Text();`

函数：**public void** saveBitmap(Bitmap bitmap)

实现将 bitmap 存储到 sdcard 中。

函数：**private** Bitmap getDiskBitmap(String
pathString)

实现将 sdcard 中的图片读出并转为 bitmap 文件。

4.3 dbHelper

实现数据库的基本方法。

后记：

关于 adb shell 的使用，ndk 的安装以及 git 等方面的知识
可以到我的博客看看，里面有讲解，地址：

<http://blog.csdn.net/buptgshengod>。

我的邮箱：

zixiyaoren@gmail.com

没看得地方可以发邮件给我。

